BHAVANA CK
1BM20CS403
CSE-4A

Program 6 : Order Database

Consider the following schema for Order Database:

SALESMAN (*Salesman_id, Name, City, Commission*)
CUSTOMER (*Customer_id, Cust_Name, City, Grade, Salesman_id*)
ORDERS (*Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id*)
Write SQL queries to

1. Count the customers with grades above Bangalore's average.

2. Find the name and numbers of all salesmen who had more than one customer.

3. List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation.)

4. Create a view that finds the salesman who has the customer with the highest order of a day.

5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

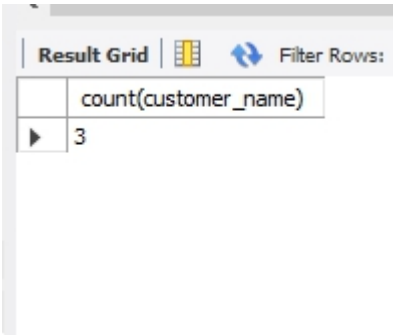
```
create database Orderdb;
use Orderdb;

create table Salesman(
salesman_id int not null,
salesman_name varchar(20) not null,
city varchar(20) not null,
commission int not null,
primary key(salesman_id)
);

create table Customer(
```

```
customer_id int not null,
customer_name varchar(20) not null,
city varchar(20) not null,
grade int not null,
salesman_id int,
primary key(customer_id),
foreign key(salesman_id)references Salesman(salesman_id) on
delete set null
);


create table Orders(
order_id int not null,
purchase_amt int not null,
order_date date not null,
customer_id int not null,
salesman_id int,
primary key(order_id),
foreign key(customer_id)references Customer(customer_id),
foreign key(salesman_id)references Salesman(salesman_id) on
delete set null
);


insert into Salesman
  values(1000,'John','Bangalore',25),
  (2000,'Ravi','Bangalore',20),
  (3000,'Kumar','Mysore',15),
  (4000,'Smith','Delhi',30),
  (5000,'Harsha','Hyderabad',15);

  insert into Customer
  values(10,'Preethi','Bangalore',100,1000),
  (11,'Vivek','Mangalore' ,300,1000),
  (12,'Bhaskar','Chennai',400,2000),
```

```
(13,'Chethan','Bangalore',200,2000),
(14,'Mamatha','Bangalore',400,3000);

insert into Orders
values(50,5000,'2017-05-04',10,1000),
(51,450,'2017-01-20',10,2000),
(52,1000,'2017-02-24',13,2000),
(53,3500,'2017-04-13',14,3000),
(54,550,'2017-03-09',12,2000);
```

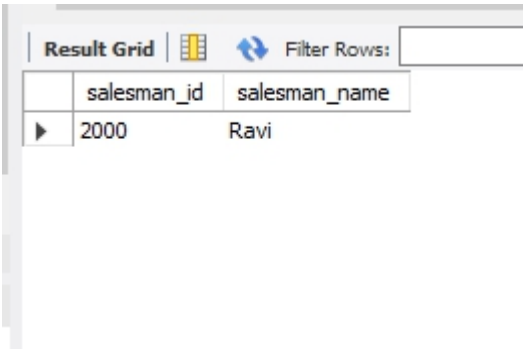
----- Count the customers with grades above Bangalore's average.

```
select count(customer_name)from Customer where grade>
(Select avg(grade) from Customer where city ='Bangalore');
```



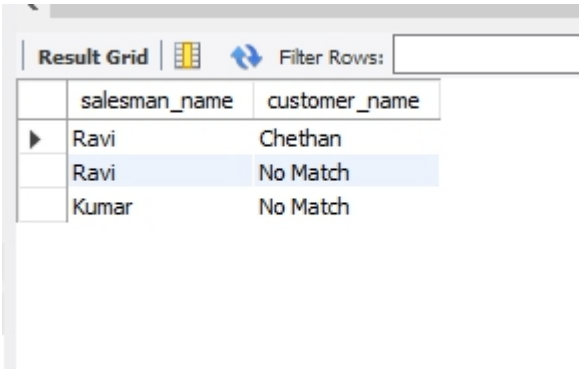
------- Find the name and numbers of all salesmen who had more than one customer.

```
select distinct c.salesman_id,s.salesman_name from
Customer c,Salesman s
where c.salesman_id=s.salesman_id
and 1<(select count(customer_id) from Customer where
salesman_id=c.salesman_id);
```



------ List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation.)

```
select s.salesman_name,c.customer_name from Salesman
s,Customer c
where s.salesman_id=c.salesman_id and c.city=s.city
union
select s.salesman_name,'No Match' from Salesman
s,Customer c
where s.salesman_id = c.salesman_id and c.city!=s.city;
```



-------Create a view that finds the salesman who has the customer with the highest order of a day.

```
create view salesman_view as
select o.order_date ,salesman_id,sum(o.purchase_amt)from
Orders o group by order_date
having sum(purchase_amt)=(select
max(sum(purchase_amt))from Customer
```

where order_date =o.order_date and salesman_id =o.salesman_id);

order_date	salesman_id	sum(o.purchase_amt)
2017-01-20	2000	450
2017-02-24	2000	1000
2017-04-13	3000	3500
2017-03-09	2000	550

------Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

```
delete from Salesmn where salesman_id=1000;
select*from Salesman;
select * from Orders;
```

Result Grid | Filter Rows: | Edit:

salesman_id	salesman_name	city	commission
2000	Ravi	Bangalore	20
3000	Kumar	Mysore	15
4000	Smith	Delhi	30
5000	Harsha	Hyderabad	15
NULL	NULL	NULL	NULL

Result Grid | Filter Rows: | Edit:

order_id	purchase_amt	order_date	customer_id	salesman_id
50	5000	2017-05-04	10	NULL
51	450	2017-01-20	10	2000
52	1000	2017-02-24	13	2000
53	3500	2017-04-13	14	3000
54	550	2017-03-09	12	2000
NULL	NULL	NULL	NULL	NULL